

**SOFTWARE VERSION DESCRIPTION  
FOR THE DII COE  
COMMON MESSAGE PROCESSOR  
Version 1.1.0.0**

**Sun Solaris 2.5.1**

**Revised April 24, 1997**

**Prepared by and for:  
Program Manager  
Common Hardware Software  
Ft. Monmouth, NJ 07703**

# TABLE OF CONTENTS

<u>SECTION</u>	<u>PAGE</u>
<b>1. SCOPE.....</b>	<b>1</b>
<b>1.1 Identification .....</b>	<b>1</b>
<b>1.2 System Overview .....</b>	<b>1</b>
<b>1.3 Document Overview .....</b>	<b>1</b>
<b>2. REFERENCED DOCUMENTS .....</b>	<b>2</b>
<b>3. VERSION DESCRIPTION .....</b>	<b>3</b>
<b>3.1 Inventory of Materials Released .....</b>	<b>3</b>
<b>3.2 Inventory of Software Contents.....</b>	<b>3</b>
<b>3.3 Changes Installed.....</b>	<b>4</b>
<b>3.4 Adaptation Data.....</b>	<b>4</b>
<b>3.5 Related Documents .....</b>	<b>8</b>
<b>3.6 Installation Instructions.....</b>	<b>8</b>
<b>3.7 Possible Problems and Known Errors.....</b>	<b>8</b>
<b>4. NOTES.....</b>	<b>8</b>

**Appendix A. Deltas**

## **1. SCOPE**

### **1.1 Identification**

This Software Version Description (SVD) applies to the Common Message Processor (CMP), Version 1.1.0.0 (non-DCE, i.e., socket based).

### **1.2 System Overview**

The CMP is the Common Operating Environment's message handling portion of the common support software suite. The CMP is a generic, table-driven United States Message Text Format (USMTF) processor that prepares, receives, analyzes, and validates USMTF message. A query interface modeled after Structured Query Language (SQL) supports the retrieval and transfer of data from USMTF messages to host system applications software. The system can analyze all USMTF message types, identifying structure and content errors.

The CMP provides tools to aid in the preparation and editing of formatted text messages such as USMTF messages. The system can accommodate all messages in the USMTF standards as well as the other agreed upon message standards. These may be customized to meet user's needs. The system also supports plain language address databases for message addresses.

The CMP provides normalization software that converts message data into a format usable by the host application software from in-coming messages. It also converts data in the format used by the host application software or databases into that of the USMTF or other message formats for out-bound messages.

The CMP MDLMAP program provides for the automatic generation of text formatted messages from information extracted from operational databases. The CMP provides for Message Journaling.

The CMP provides an User's Interface (CUI) which allows the user access to various capabilities of the CMP. The CUI provides the user with a client based connection that can be utilized with the Journaling server to provide a distributed method of monitoring all incoming and outgoing messages. Additionally, the CUI provides the user with the capability to interactively create/edit messages by accessing the CMP Message Generation /Edit capability. Finally, the CUI provides a means to filter the incoming and outgoing windows of the CUI. This filtering capability allows the users to filter on FROM, TO, DTG, Precedence, Classification, etc., and to perform various sorts on the message fields.

### **1.3 Document Overview**

The purpose of this SVD is to identify and describe the CMP. Section 2 is a list of the documents referenced in this manual. Section 3 describes version 1.1.0.0 of the CMP Software.

## **2. REFERENCED DOCUMENTS**

- Application Programming Interface (API) for the DII COE CMP, Version 1.1.0.0 on the Sun Solaris, dated 24 April 1997
- Software Requirements Specification (SRS) for the DII COE CMP, Version 1.1.0.0 on the Sun Solaris, dated 24 April 1997
- Software User's Manual (SUM) for the DII COE CMP, Version 1.1.0.0, dated 24 April 1997
- Software Test Description (STD) for the DII COE CMP, Version 1.1.0.0, dated 24 April 1997
- Software Test Report (STR) for the DII COE CMP, Version 1.1.0.0, dated 24 April 1997
- Software Test Plan (STP) for the CMP Version 1.1.0.0, dated 24 April 1997

### 3. VERSION DESCRIPTION

#### 3.1 Inventory of Materials Released

The COE CMP Software, Version 1.1.0.0 is released on 8 mm Digital Audio Tape (DAT) in Unix "tar" format.

#### 3.2 Inventory of Software Contents

The COE CMP Software, version 1.1.0.0 consists of the following files:

Directory structure:

CUI/  
 autofill/ bin/ config/ msgs/ bitmap/ bitmaps/ system/ state/  
 SegDescrip/ Scripts/

DISC  
 bin/ state/ inbound/ outbound/ data/ bitmap/ LIB/ discarded/  
 SegDescrip/ Scripts/

JMAPS/  
 config/ mio/ mioref/ badFormatMsgs/ demo/ duplicateMsgs/ etc/ bin/  
 doc/ incompleteMsgs/ storage/ data/ skel/ bitmap/ SegDescrip/ Scripts/

JMPS/  
 bin/ lib/ man/ spool/ SegDescrip/ Scripts/

JOURNALING/  
 inbound/\* outbound/\* bin/ doc/ bitmap/ data/ SegDescrip/ Scripts/

NORMALIZATION/  
 datafiles/ include/ obj/ bin/ environment/ include\_tem/ user\_manual/  
 data/ examples/ lib/ SegDescrip/ Scripts/

MTFVMFParser/  
 lib/ bin/ state/ SegDescrip/ Scripts/

DTJMPS/  
 bin/ data/ SegDescrip/ Scripts/

DTJMAPS/  
 bin/ data/ SegDescrip/ Scripts/

\* The inbound and outbound directories for JOURNALING are created during installation.

Executable programs are found in the bin directories. Message data tables for USMTF messages are found in the DTJMAPS/data directory for JMAPS and the DTJMPS/data directory for JMPS.

SegDescrip contains release notes, version notes, and DII COE configuration setup parameters.

Scripts contains installation and runtime scripts.

### 3.3 Changes Installed

Added changes to CMP User's Interface. Corrected STRs. Created another segment, MTFVMFParser. Changed the CMP segment layout to a single aggregate segment, with the JOURNALING segment as the parent segment.

### 3.4 Adaptation Data

Variable Message Format:

- The VMF library provides the capability to successfully Encode and Decode VMF messages as defined in the VMF TIDP CH3 January 26, 1996.
- The Library provides the following APIs:

```

/*****
/
/*                               *
/
/* routine: unsigned32 vmfhdr2mtf()           *
/
/* Translate vmf bit-oriented header message to mtf ascii msg           *
/
/* Input:                               *
/
/*  unsigned char *vmf_msg:           *
/
/*   VMF Message Buffer           *
/
/*                               *
/
/* Output:                               *
/
/*  char *header:           *
/
/*   A Pre-allocated buffer to store the ASCII mtf header message.   *
/

```

```

/*          *
/
/* Returns: unsigned32          *
/
/*   Number of bytes parsed in vmf msg.          *
/
/*          *
/
/*****
/
unsigned32 vmfhdr2mtf( unsigned char *vmf_msg, char *header )

/*****
/
/*          *
/
/* routine: unsigned32 vmf2mtf()          *
/
/* Translate vmf bit-oriented message to mtf ascii msg          *
/
/* Input:          *
/
/*   unsigned char *vmf_msg:          *
/
/*   VMF Message Buffer          *
/
/*   char *tfilename:          *
/
/*   The file containing the message table for parsing this message.          *
/
/*          *
/
/* Output:          *
/
/*   char *mtf_msg:          *
/
/*   A Pre-allocated buffer to store the ASCII mtf message.          *
/
/*          *
/
/* Returns: unsigned32          *
/
/*   Number of bytes in vmf_msg.          *
/
/*          *

```

```

/
/*****
/
unsigned32 vmf2mtf( unsigned char *vmf_msg, char *filename, char *mtf_msg )

/*****
/
/*
/*
/* routine: unsigned32 mtfhdr2vmf()
/*
/* Translate mtf ascii msg to vmf bit-oriented message
/*
/* Input:
/*
/* char *header:
/*
/* A buffer containing the ASCII mtf header message.
/*
/*
/* Output:
/*
/* A Pre-allocated VMF Message Buffer
/*
/*
/* Returns: unsigned32
/*
/* If success:
/*
/* Number of bytes in vmf_msg.
/*
/* If NOT success:
/*
/* VMF_ERROR.
/*
/*
/*****
/

unsigned32 mtfhdr2vmf( char *header, unsigned char *vmf_msg )

```

```

/*****
/
/*          *
/
/* routine: unsigned32 mtf2vmf()          *
/
/* Translate mtf ascii msg to vmf bit-oriented message          *
/
/* Input:          *
/
/* char *mtf_msg:          *
/
/*   A buffer containing the ASCII mtf message.          *
/
/* char *tfilename:          *
/
/*   The file containing the message table for parsing this message.  *
/
/*          *
/
/* Output:          *
/
/* unsigned char *vmf_msg:          *
/
/*   A Pre-allocated VMF Message Buffer          *
/
/*          *
/
/* Returns: unsigned32          *
/
/* If success:          *
/
/*   Number of bytes in vmf_msg.          *
/
/* If NOT success:          *
/
/*   VMF_ERROR.          *
/
/*          *
/
/*****
/
unsigned32 mtf2vmf( char *mtf_msg, char *tfilename, unsigned char *vmf_msg )

```

- These APIs will provide a user with the ability to convert from a VMF message to an MTF representation.
- Additionally, PM CHS provides VMF translator library to convert from VMF to a C Data structure equivalent, and to convert from VMF to an ADA data structure equivalent. To obtain these libraries, please contact: Win Parks, PM CHS, 908-427-4991. For Technical support, please contact: Robert D. Gerardi, Technical Manager CMP, Unixpros Inc., 908-389-3295 x546.

### **3.5 Related Documents**

None.

### **3.6 Installation Instructions**

The installation instructions are provided in the Software User's Manual (SUM) for the DII COE CMP.

### **3.7 POSSIBLE PROBLEMS AND KNOWN ERRORS**

Known problems for all versions are documented in the Software Test Report (STR) for the DII COE CMP.

### **4.0 NOTES**

None

**APPENDIX A**

Deltas:

1. CUI/bin/cui\_sock\_fb  
CUI/bin/cui\_trarc\_fb  
CUI/bin/cui\_hpdce\_fb - The CUI includes the VMF and ACP123 Header selection and input.
2. MTFVMFParser - Added this segment to allow other mission applications to access the CMP BOM\_to\_COM translator.
3. JOURNALING/SegDescrip/SegName - Added all of the child segments. Removed the SEGMENT descriptor.
4. NORMALIZATION/SegDescrip/SegName - Added the JOURNALING entry as its parent segment. Removed the SEGMENT descriptor.
5. DTJMPS/SegDescrip/SegName - Added the JOURNALING entry as its parent segment. Removed the SEGMENT descriptor.
6. DTJMAPS/SegDescrip/SegName - Added the JOURNALING entry as its parent segment. Removed the SEGMENT descriptor.
7. JMPS/SegDescrip/SegName - Added the JOURNALING entry as its parent segment. Removed the SEGMENT descriptor.
8. JMAPS/SegDescrip/SegName - Added the JOURNALING entry as its parent segment. Removed the SEGMENT descriptor.
9. CUI/SegDescrip/SegName - Added the JOURNALING entry as its parent segment. Removed the SEGMENT descriptor.
10. DISC/SegDescrip/SegName - Added the JOURNALING entry as its parent segment. Removed the SEGMENT descriptor.